

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Group Art Unit: 2195
Examiner: Camquy Truong

March 20, 2006

Customer Assignment No. 027516
Serial No.: 10/038,547
Filed: October 19, 2001
In re Application of: James W. Willen et al.
Title: Operating System Scheduler/Dispatcher with
Randomized Resource Allocation and User
Manipulable Weightings
Docket No.: RA-5368

Amendment

via email transmission
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Madam,

This amendment is being submitted in response to the telephone interview conducted March 20, 2006 in the subject case. Applicants' Representative thanks the Examiner for conducting that interview. This amendment is being submitted to correct formality issues only.

Claim Amendments

It is requested that the following amendments be entered into the Claims:

5. A two-stage lottery program for a dispatcher program that dispatches tasks within an operating system of a computer system, the computer system

supporting at least two classes of said tasks, each of said classes including at least two levels of said tasks, said dispatcher program to determine which of said tasks will be assigned to a next available instruction processor resource, said two-stage lottery program comprising:

a random number generator and selection program for generating in a first lottery stage a first random number for selecting one of said at least two classes, and for generating in a second lottery stage a second random number for selecting one of said at least two levels within said selected class; and

a transfer program for assigning said next available instruction processor resource to execute a task assigned to said selected one of said at least two levels.

6. The two-stage lottery program of claim 5 further comprising:

a level switching routine to handle a failure by said transfer program to find a task on said selected one of said at least two levels and to assign said next available instruction processor resource to a task associated with a different one of said at least two levels.

7. The two-stage lottery program of claim 5 wherein any one of said at least two levels is two times more likely to be selected than a next lower one of said at least two levels.

8. The two-stage lottery program of claim 7 wherein tasks within each level of said at least two levels will be assigned to said next available instruction processor resource for a same amount of time as other tasks in said level.

9. The two-stage lottery program of claim 5 wherein each of said tasks is respectively associated with an amount of time said each of said tasks may

continuously execute on said next available instruction processor resource, and wherein said two-stage lottery program employs a quantum bias routine comprising:

- a data capture routine for determining how much of said associated amount of time said task executes on said next available instruction processor resource before returning control to said dispatcher; and

- a bias adjustment routine for adjusting said amount of time associated with said task based on how long said task last executed on said next available instruction processor resource.

10. The two-stage lottery program of claim 9 wherein said bias adjustment routine does not adjust said amount of time associated with said task if execution of said task was interrupted by an interrupt.

12. A computer system having a quantum timer to allow an instruction processor resource to be assigned to process one or more tasks, the computer system also having a dispatcher program whereby all of said one or more tasks are identified as being members of classes, said dispatcher program comprising:

- a) a scheduler to determine for how long, and to which of said one or more tasks, said instruction processor resource will be next assigned; and
- b) a scheduler queue from which said one or more tasks may be assigned to said instruction processor resource, wherein

- said scheduler provides a two stage lottery, a first stage using a lottery process to select one of said classes and a second stage using a second lottery process to select from said selected class a level, wherein said scheduler selects a task that will next be assigned to said instruction processor resource from said one or more tasks that are members of said selected level of said selected class.

13. The computer system set forth in claim 12 wherein said first stage chooses a class randomly from among all said classes using a bias settable by a

user.

14. The computer system set forth in claim 12 wherein the number of said classes is selectable by a user.

15. The computer system as set forth in claim 12, wherein any of said one or more tasks included within one or more predetermined classes are assigned said instruction processor resource prior to said scheduler providing said two stage lottery.

16. The computer system as set forth in claim 12, wherein if the scheduler selects one of said classes that does not have any of said one or more tasks as members, said scheduler chooses another one of said classes.

17. The computer system as set forth in claim 12, wherein if none of the classes has any of said one or more tasks as members, said scheduler selects a low priority level operating system task.

18. The computer system as set forth in claim 12, wherein said second stage lottery comprises:

a random number generator and selection program for generating a random number and for selecting said selected level within said selected class based upon said random number; and

a transfer program for causing said instruction processor resource to begin execution of one of said one or more tasks that is assigned to said selected level.

19. A method for use by a dispatcher of a computer system for selecting one of multiple tasks to provide with an available instruction processor resource, wherein said one of said multiple tasks may be selected from one or more

classes, each of which may have one or more priority levels, wherein said method comprises:

determining whether there are any of said multiple tasks within said priority levels of said classes that are above a second stage lottery line, and if so assigning a first of said multiple tasks to said available instruction processor resource; but if not,

and if there is only one said priority level having any of said multiple tasks, assigning a first of said any multiple tasks from said only one priority level to said available instruction processor resource; else,

running a first stage of a two-stage lottery algorithm to select one of said one or more classes, running a second stage of said two-stage lottery algorithm to select a priority level from said selected class, and selecting said selected one of said multiple tasks from said selected priority level to provide to said available instruction processor resource.

20. A method for use by a dispatcher in an operating system of a computer system for selecting one of multiple tasks to assign to an available instruction processor resource, said selected task being selected from one or more classes of tasks, each having one or more priority levels, wherein said method comprises:

if there is only one of said priority levels having any of said multiple tasks, assigning a first of said any of said multiple tasks to said available instruction processor resource, else,

running a first lottery stage of a two-stage lottery algorithm to select one of said one or more classes, and running a second lottery stage of said two-stage lottery algorithm to select from said selected one of said classes a priority level from which to select said selected task to assign to said available instruction processor resource.

21. The method of claim 20 further comprising:
- respectively associating each of said multiple tasks with an amount of time said each task is entitled to be assigned to said instruction processor resource;
 - placing each of said multiple tasks into any one of said priority levels based upon said associated amount of time; and
 - moving said selected task from a current priority level to a different priority level based on how much of said associated amount of time said selected task used a last time said selected task was assigned to said available instruction processor resource.
22. The method of claim 21, said moving step further comprising:
- if said selected task did not complete execution last time said selected task was assigned to said available resource, moving said selected task from said current priority level to a next lower priority level and increasing said associated amount of time by a predetermined amount,
 - if said selected task used less than all, but more than a predetermined portion, of said associated amount of time last time said selected task was assigned to said available resource, leaving said priority level and said associated amount of time the same, and
 - if said selected task used less than said predetermined portion of said associated amount of time last time said selected task was assigned to said available resource, moving said selected task from said current priority level to a next higher priority level and decreasing said associated amount of time by a predetermined amount.
23. The method of claim 21 wherein each of said multiple tasks is represented by a task pointer stored on a scheduler queue.

24. The method of claim 21 wherein said first stage lottery selects which of said classes of tasks has tasks assignable to said available instruction processor resource.

25. The method of claim 21, further comprising:

if any of said multiple tasks are in classes above a lottery line, providing said available instruction processor resource to said any of said multiple tasks in said classes above said lottery line in a predetermined priority order, and otherwise

using said first stage lottery to select which of said classes of tasks have tasks assignable to said available instruction processor resource.

26. A dispatcher system for use within a computer system, said computer system having an instruction processor resource and a scheduler for storing pointers to tasks, each of said tasks to be assigned to said instruction processor resource for an amount of time associated with said each task, said dispatcher system comprising:

a first stage of a two stage lottery system for using a random number generator to select one of multiple classes of tasks; and

a second stage of said two stage lottery system for selecting a task from said selected one of said multiple classes, whereby said selected task is assigned to said instruction processor resource for said amount of time associated with said selected task.